SOA Governance: Better Now Than Never

By Alex Nubla

The year was 2004. **John** was the CTO for an IBM ISV where they piloted a supply chain application. They used the SOAP engine inside the IBM Websphere Application Server, with Enterprise Java Bean for the front end and connections to i5 RPG applications using WSDL.

The idea behind this was point-to-point integration, the kind where Web Services (WS) and consuming applications are designed at the same time. These were easy to create. But as their business changed and more clients requested Statements of Work (SOW), those point-to-point WS became obsolete. Worse, since these WS were easy to build, developers had a blast creating a plethora of services. The Quality of Service was in various states and some not well thought of.

By 2007 the company had created way too many WS with little or no long-term thinking. No one was sure which applications were using which WS, and with so many SOW created, no one knew the history and ownership.

SOA is great, but unless it is managed, it is just another IT process that can be chaotic. John contacted Gartner and they recommended putting into place a SOA Center of Excellence: to act as a focal point; to establish reference architecture; to review newly created WS; to communicate & enforce the standards; and to encourage a higher degree of service reuse. Implementations are typically coordinated with clients, and unless strict governance exists, different groups or individual developers will make different decisions, increasing diversity and inhibiting reuse.

Things had been out of control for some time. John realized that the current mode of operation, with frequent ad hoc WS creation could not be continuously sustained, and could not support the numerous SOW requests. John finally understood the difference between WS and SOA. It is easy to create WS; but it takes discipline to deliver meaningful and serious value from them. John had been using a spreadsheet to track WS and all their dependencies. This was no longer sufficient. Many clients were asking for more details about the various services. John had a hard time keeping the spreadsheet up to date and it was error prone. The idea of having a registry and repository for SOA is exactly what was needed. How could John encourage reuse if no one could find information about what services might be available for reuse?

Tim Everett, the CFO of National Parts Distributor (NPD), approached John about modifying their Web Services enabled Warehouse Management System (WMS). NPD required various inventory inquiry services which needed to be consumed by .NET clients. According to Tim, for time-to-market reasons, the WS need to integrate with third party services to locate parts or preferred distribution centers.

John realized his strategy of point-to-point WS was over. A whole new chapter of flexible relationships between service consumer and provider is apparent. This, together with understanding the purpose of each WS and dependencies were vital. NPD is not the only customer requesting such changes.

John pondered on how to prioritize and execute a solution:

- How can we deliver more services to our customers quickly?
- How can we prioritize the services that needed to be built?
- How can we encourage services reuse so that cost and time required to get applications running can be reduced?
- How can we control the Quality of Service being implemented?
- How can we identify and improve those hard-to-integrate services?
- How can we identify best practices and pattern for better services?
- How can best practices be enforced?
- How can we get a handle on the entire

Lifecycle Stage:	Design Time	Run Time	Change Time
Stakeholder:	Architect, Developer, COE, QA / Testers	IT Operations	Business Users, IT Administrator
Policy Store:	Registry / Repository		
Policies:	Change Management, Publication, Policy Lifecycle	Core Security, Service Usage, Quality of Service	Auditing
Policy Enforcement:	Registry Repository	Message Transport	Management System

Figure 1. Policy Enforcement Points (PEP) for each Lifecycle Stage

service inventory, the infrastructure required to run on, and the other services or composite applications that depend on them?

Without answers to these questions, John knew he couldn't deliver SOA with the promise of agility. Then he recalled reading about SOA Governance.

SOA Governance is a misunderstood term. Some people use the term to mean service lifecycle governance—governing from creation to implementation. Others coined it as applying runtime policies to services. But is there more to SOA Governance than this?

SOA Governance is a concept used for activities related to exercising control over services in SOA. It is an emerging concept used to address management issues that are caused by the *loose coupling* of services in SOA. SOA Governance can be seen as an overlay on IT Governance, but often has a more organizational focus than IT Governance, when services represent business activities.

To achieve SOA Governance, John addressed the following:

1. He collected three years worth of artifacts related to services, and investigated the end-to-end lifecycle of these assets. Artifacts include XML, WSDL, XLS and multitude of documents from Excel and Word.
2. He established initial policies, standards, and key measurements related to the lifecycle of services and composite applications.
3. He reviewed SOA Governing mechanism software in the market that enforces policies, decisions, and processes around the end-to-end lifecycle of these artifacts.

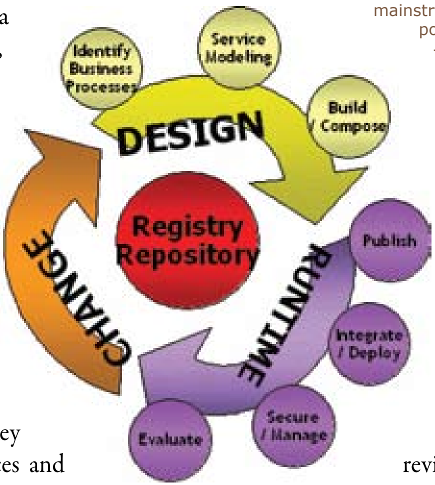

Figure 2.

Exploring SOA Governance Solutions

Software system mechanisms play a major role in the foundation of SOA governance to enforce and automate policies across the services lifecycle. Two of the main components of this system are:

- A Registry, which acts a central index of business services
- A Repository, for storing policies and other metadata related to the governance of these services

By themselves, these components are insufficient. The registry and repository must be fully interoperable with each other and with other SOA artifacts. They must form a comprehensive system that manages the entire SOA lifecycle.

Define Policies and Procedures

John's initial step is to define the policies for the company. This is done in conjunction with the SOA Center of Excellence to help communicate policies and other SOA related decisions. Policies include both business and technical requirements, and help create a common source of information and processes. You should be able to answer the following questions:

- Which policy should you implement first? What policies are needed now?
- Who in your organization is responsible for creating policies?
- How will you create these policies? How will you communicate them?
- Which policies can be automated?
- How will people within your organization discover policies?
- What tools will people use to follow policies?
- How will management get visibility into the policy compliances?
- What actions should be taken for policy violations?

The first step of SOA Governance is to define your framework. The framework tells the organization what to consider during policy creation, communication and enforcement relevant to the project. This governance framework then becomes your outline. Take an iterative, step-by-step or phased approach to your governance. Your initial attempt may include simple policy documentation, but your next iteration includes detail definitions and how you plan to enforce such policies. As your organization moves up the maturity model, you can create more enterprise-based policies and processes around shareable services. Governance becomes more important and the scope widens as SOA becomes more mainstream and reuse increases. Establish SOA goals, standards, policies and procedures proportionate to your SOA maturity.

Establish SOA Baseline

John can evaluate the business impact of introducing new or changing policies against current artifacts and services already published in the registry and repository. SOA Governance involves the enactment of these components—checking to make sure the artifacts follow policies, publishing reports on results & business impacts, and dynamically associating each artifact with its location. Having an SOA baseline allows John to see the quality of existing artifacts and highlights areas that need review. For the first time, John has a dashboard view that he can share with senior management to provide visibility on the current state of their SOA.

Arm yourself with information about where to start; which policies have the highest impact on business and which are out of compliance. Focus in improving services that do not comply with policies and have the highest impact on business. Gradually bring existing services to comply with your policies.

Establish SOA Roles and Responsibilities

John formed an SOA Center of Excellence (COE) based on the recommendation from Gartner. The SOA COE became the governing body and single point of coordination for SOA. Clients now have a focal point to call when they require more details on various Web services. John assigned subject matter experts, business analysts and IT architects to the COE, and assigned each with a clear set of responsibilities. John also appointed a Manager to chair the COE.

SOA Center of Excellence provides the perfect opportunity to see how business objectives can be articulated in SOA. The roles and responsibilities include:

- Acting as the SOA focal point
- Communicating and enforcing standards
- Helping to establish a reference architecture
- Reviewing newly created services
- Encouraging a high degree of services reuse
- Helping to setup policies and procedures
- Suggesting corrective actions when a specific process is broken
- Defining the IT infrastructure to support SOA
- Developing clear guidelines for SOA projects
- Discovering how to implement SOA and Web services to help increase competitive advantage
- Adopting proven best practices to optimize SOA

Activity	Deliverable	Stage
Business Service Analysis	An understanding of data entities and process steps that drive the need for the creation of a service	Planning
Service Partitioning	An understanding of the different levels of services (data level, orchestration, composition, management) needed to meet the needs of the business, what each service will do. This drives the definition of business events and documents.	Planning, Design
Event and Schema Design	The plan for the behavior of the services that meets the operational, informational, and business process needs of your organization. Behavior is often described as a protocol, but it can include service level expectations, exception management and compensation definition.	Planning, Design
Security Policy Creation & Management	These are the set of standards for how services will be secured, what level of authorization is needed for services of different types, how network boundaries will affect the access to different forms, levels, and types of data.	Planning
Operational Policy Creation & Management	These are the set of standards for how services will be constructed so that they can be seen, tracked, managed, audited, and monitored.	Planning
Policy Enforcement	Automated application of policies to services running in the network	Implementation, Support
Service Monitoring	Automated monitoring, logging, and tracking of service calls to insure that service levels are maintained and to aid in debugging and exception handling.	Implementation, Support
Rogue service discovery	Automated discovery of services running in the network to capture services that may offer uncontrolled functionality, backdoor access, and audit gaps.	Support
Service Registry and Repository	Tools for sharing information about services, both with consuming applications and with the people who create or use them.	Planning, Design, Build, Implementation, Support
SOA Project Compliance	A process for insuring that projects funded in corporate IT departments actually consume or deliver the services needed by the enterprise.	Planning, Design, Build

Figure 3. The highlighted rows are areas often covered by SOA Governance System Mechanisms. While these are important "Strategic" governances, they are about 20% of the whole SOA Governance.

Continuous Governance

Now that the SOA baseline and COE exist, John can enforce the policies accordingly to any artifacts published in the registry and repository, ensuring continuous governance over newly created services and enabling changes to existing ones.

As part of policy enforcement, John can configure criteria of publication acceptance and rejection. In certain occasions, he can allow publication even though the artifact is not compliant, and can mark through the metadata that this artifact should not be exposed as Production Quality. John is aware there are exceptions to every rule—as artifacts are checked for compliance users may want to request exceptions from complying with these policies. Exceptions must be managed as well.

SOA Governance Foundation

The foundation of SOA Governance is the ability to enforce and automate policies across the SOA lifecycle. A set of mechanisms can enable the automation and enforcement. To do this, we need a place to store policies (Policy Repository) and a place to enforce them (Policy Enforcement Point). These two locations represent where a policy "lives" and where it "goes to work."

The SOA lifecycle consists of three stages:

1. Design Time
2. Run Time
3. Change Time

Regardless of lifecycle stage, the Registry & Repository is the Policy Store for all stages. Having a single context for policies enables SOA lifecycle management of policies and a means to establish enterprise-wide policies.

However the Policy Enforcement Point (PEP) changes with the lifecycle stages. As illustrated in **Figure 1**, during Design Time – the Registry / Repository itself is the PEP. During Run Time – the Message Transport system (your ESB) is the PEP. And finally during Change Time – the Management and Security system is the PEP.

Design Time

- Architect designs a service contract or plans a service implementation
- Developer searches for an existing service before building a new one
- Developer requests that the SOA COE approves the creation of the service
- Tester simulates a new service to execute a test plan

During Design Time, policies such as namespace validation, schema validation, interoperability validation, approvals, document access control, audits, and resource utilization need to be enforced. If the Policy Repository is combined with the Service Registry, then typical best practice is for the SOA Registry / Repository system to also serve as the enforcement point.

Run Time

- IT Operations monitor a business process
- IT Operations confirm compliance of a service with policies (or SLA)
- Operations Manager monitors level of service reuse
- Operations Manager handles service exceptions

The Run Time system is responsible for Message Transport. Enterprise Service Bus (ESB) is typically used. The ESB brokers transactions between service providers and consumers and frequently handles functions such as data transformation, reliable messaging & message queuing, security, and other operations. In a Web service, the emphasis is on supporting SOAP and extensions like WS-Security. These systems can act as Runtime PEP.

Change Time

- Business Analyst plans a change within certain business process
- IT Administrator adjusts Quality of Service requirements for a service

Change Time governance is the act of managing services through the cycle of change. Since most services will be modified through time, this is an essential component of long-term governance. During Change Time, users connected to services, taxonomies, or policies receive notification to approve, review, or recognize when any of those assets are modified. Furthermore, Change Time governance enables policies to dictate whether a service or even another policy can change—and if so, who approves it. If a change is planned (e.g. version upgrade, service decommission, etc.) then the event is recorded and described to allow dependent parties to adjust their processing.

SOA Lifecycle within SDLC

The whole point of SOA is to create an agile environment, making it easier to build a fully integrated environment from the get-go. (See **Figure 3**.) This is our goal. If our services do not allow us to build service oriented applications, then we just wasted money and time. SOA governance is about making sure we don't waste time and money by building services we don't need, or failing to build services we need.

Bottom Line

An ungoverned SOA can often lead to unintended consequences, reversing the cycle and causing SOA to add cost and disrupt processes. There is no "one size fits all" SOA Strategy. Thus, there is no such thing as a single set of policies and procedures that constitute governance with SOA. We should not forego SOA because of the risk, but rather define strategy for SOA that builds governance. The cost of ungoverned SOA can cause:

- Lack of reuse by compromising trust, unpredictable quality, and performance issues
- Process disruption by publishing services that do not conform to standards or SLA, or by failure to assess the impact of change
- Escalations in support costs through an onslaught of help-desk and service calls due to service issues and outages
- Lack of interoperability resulting in silos of business services, and perpetuating the same issues of traditional "tightly coupled" architecture
- Non-compliance, by failing to associate key policies with the services that have implications for industry or governmental regulations
- Security breaches by allowing arbitrary data access
- Overall SOA failure by allowing chaos, perpetuating a "Garbage in, Garbage out" environment

Governance is not an option for SOA—it is necessary. SOA Governance requires more than a registry and repository. It requires an integrated solution that provides support for all SOA stakeholders throughout the SOA lifecycle.

***Alex Nubla** is a subject matter expert specializing in Service Oriented Architecture. He has served in a variety of technical and business roles across a broad range of industries including Banking, Finance, Supply Chain, and Health Care, to name a few. Alex's current post is as an SOA Strategic Lead and a Director for Health Net Inc.*